

## Using OWL-VisMod through a decision-making process for reusing OWL ontologies

Francisco J. García-Peñalvo<sup>a\*</sup>, Patricia Ordóñez de Pablos<sup>b</sup>, Juan García<sup>a</sup> and Roberto Therón<sup>a</sup>

<sup>a</sup>Computer Science Department, Universidad de Salamanca, Salamanca, Spain; <sup>b</sup>Department of Business Administration and Accountability, University of Oviedo, Oviedo, Spain

(Received 20 April 2012; final version received 3 July 2012)

Knowledge representation (KR) can be defined as a set of ontological commitments, provided with the capabilities of performing inference. The knowledge can be represented using an ontology, which provides a shared insight into a certain domain. The use of ontologies to represent knowledge also allows interoperability among knowledge-based systems. The process of building ontologies can be tedious and sometimes exhaustive. A possible solution in order to avoid this problem would be to reuse the ontologies previously created by others. This paper describes a case study of reusability using OWL-VisMod, a tool designed for developing ontological engineering based on visual conceptual modelling for OWL ontologies. A workflow performed with OWL-VisMod is described; including a decision-making process in order to decide whether or not it could be desirable to reuse an ontology, according to the requirements of a certain project.

**Keywords:** knowledge representation; ontology reusability process; information visualisation

### 1. Introduction

Knowledge representation (KR) is an artificial intelligence area where the fundamental goal is to represent knowledge in a way that facilitates inference. Generally speaking, some kind of logic is used both to supply formal semantics of how reasoning functions apply to symbols in the domain of discourse, as well as how to supply operators such as quantifiers, modal operators, etc. that, along with an interpretation theory, give meaning to the sentences in the logic. Recent developments in KR have been driven by the Semantic Web and especially by ontologies, providing a great variety of definitions on knowledge representation; one of these definitions is that KR is a set of Ontological Commitments (Davis *et al.* 1993).

Knowledge-based systems have special requirements for interoperability; they operate on and communicate using statements in a formal knowledge representation (Gruber 1993, Zhao *et al.* 2012). They send queries and provide answers with diverse systems, performing knowledge exchange. Ontologies are used to describe ontological commitments for a set of agents so that they can communicate about a domain of discourse without necessarily operating on a globally shared theory. It is said that an agent commits to an ontology if its observable actions are consistent with the definitions in the ontology. Even when an ontology serves a different purpose than a knowledge base (Gruber 1993), it can be used as a knowledge base in

diverse and specific areas and approaches, such as (Guarino and Giarretta 1995, Marcondes *et al.* 2008, Marcondes and Rocha 2009).

Ontologies are explicit representations of domain concepts; they provide the basic structure or armature around which knowledge bases can be built. Each ontology is a system of concepts and their relationships, in which all concepts are defined and interpreted in a declarative way. Typically, an ontology consists of a finite list of terms and the relationships between these terms. The terms denote important concepts (classes of objects) of the domain with a hierarchy. Apart from subclass relationships, ontologies may include information such as properties, value restrictions, disjoint statements and specifications of logical relationships between objects.

Ontologies have been widely used in the context of Knowledge Management Systems (Bera 2007); they represent a crucial aspect of the semantic technologies, even from the point of view of the industry. Currently, the diversity of these semantic technologies has been proposed, ranging from the Information Technology platforms such as RDi-Advise (Colomo-Palacios *et al.* 2010) or biological platforms (Stevens *et al.* 2007) to the analysis of social networks (García-Crespo *et al.* 2010).

OWL-VisMod (<http://www.analiticavisual.com/juan/OWL-VisMod.html>) aims to contribute to the development of ontological engineering, the branch of

---

\*Corresponding author. Email: [fgarcia@usal.es](mailto:fgarcia@usal.es)

knowledge engineering that exploits the formal principles to build ontologies. The main purpose behind OWL-VisMod is to provide users with a tool to support the development, creation, management, maintenance and reusability of OWL ontologies for knowledge-based systems (García *et al.* 2011a,b). The usability of OWL-VisMod has been evaluated by means of an empirical study, with good results (García *et al.* 2011c).

This paper is focused on providing a step-by-step guideline in the process of reusability of an ontology using OWL-VisMod, which fully supports this workflow. In this context, the term reusability can be defined as a collection of principles and heuristics for the creation and evolution of diverse systems, especially but not restricted to software systems (Prieto-Diaz and Arango 1991). Reuse is the reapplication of a variety of kinds of knowledge about one system to another similar system to reduce the effort of development and maintenance of that other system (Biggerstaff and Perlis 1989). As this approach of reusing software components has been widely used, the reuse of ontologies refers in a broad perspective, to reused knowledge that may consist, at a first level, of ontologies themselves, or more extensively, of knowledge that is founded in ontologies (Biris 2003). Since ontologies have been proposed as a means of specification that enables knowledge reuse and sharing, they provide a unifying conceptual vocabulary for representation of data sets.

The main problem with reusing ontologies is the semantic binding (Biggerstaff and Perlis 1989, Karlsson 1995); which refers to the identification of concepts that are semantically equivalent to the concepts in the new domain. This semantic binding process has been poorly supported by existing tools. Although Ontolingua Server (<http://www.ksl.stanford.edu/software/ontolingua/>) stands out as the main proposal regarding sharing and reusing ontologies, despite the fact that this environment provides a distributed and collaborative platform, the semantic binding has to be performed by the user by manually checking the concepts of the ontologies, in order to decide whether or not they are semantically equivalent to what the user is really looking for.

According to the analysis described in García-Peñalvo *et al.* (2011, 2012), there are neither methodologies nor workflows that support reusability of ontologies. Aimed at filling that void, in this paper a workflow is proposed and a case study is described according to it. The workflow has been defined based on the main phases of two of the most important methodologies for building an ontology, such as Methontology (Fernández *et al.* 1997, Gómez-Pérez *et al.* 2003) and Developing Ontology-Guided

Mediation for Agents (DOGMA), a framework to formally develop the ontological engineering (Jarrar and Meersman 2002, Spyns *et al.* 2002). The main philosophy of the DOGMA methodology is to reuse ontologies, because they are considered as scalable and shared resources that let to reuse the knowledge (Jarrar and Meersman 2008).

The remainder of the paper is organised as follows: section two provides a summary of the most important and currently available tools for modelling and visualising ontologies, pointing out their disadvantages as compared to OWL-VisMod. In the third section, there is a brief description of the tool as well as a case study using OWL-VisMod. This case study is described according to a workflow that involves various phases such as the analysis of the taxonomy, properties or coupling, and it finishes with a take of a decision according to the analysis. Finally, the fourth section discusses the conclusions.

## 2. Related work

Visual language is one of the oldest forms of knowledge representation and predates conventional written language by almost 25,000 years (Chen 2006). Visual representations are effective because they tap into the capabilities of the powerful and highly parallel human visual system (Kerren *et al.* 2008). Diagrams can convey information more concisely and precisely than ordinary language. Information represented visually is also more likely to be remembered due to the superiority of the effects of a picture (Moody 2009). A visual notation (or visual language, graphical notation, diagramming notation) consists of a set of graphical symbols (visual vocabulary), a set of compositional rules (visual grammar) and definitions of the meaning of each symbol (visual semantics). Together, the visual vocabulary and visual grammar form the visual (or concrete) syntax. Graphical symbols are used to symbolise (perceptually represent) semantic constructs, typically defined by a metamodel.

The visual language has evolved to a great variety of visualisations and visual techniques focused on specific aspects to analyse, in almost all the possible fields. The modelling of ontologies is not the exception, since various tools have been proposed to represent ontologies in a visual manner. The most important commercial tools currently available are: SemanticWorks (<http://www.altova.com/semanticworks/owl-editor.html>), TopBraidComposer (<http://www.topquadrant.com/>) and IODT (<http://www.alphaworks.ibm.com/tech/semanticstk>). Some of these tools offer a free version but with reduced capabilities. On the other hand, Protégé is currently the most widely used tool for editing ontologies, as it is free to download and it is

also very robust. Below, a brief description of each tool is provided, pointing out their disadvantages.

An interesting tool based on the use of conceptual maps is SemanticWorks. It is a commercial tool designed to edit RDF documents in a GUI and check its syntax, as well as design RDF schema and OWL ontologies using a graphical design view. It is also able to check the syntax and semantics of OWL Lite, OWL DL ontologies and export ontologies in the RDF/XML and N-Triples formats. The modeller uses a representation based on expandable square boxes for properties and classes; this representation is intended to convey complex conceptual knowledge bases in a clear and understandable way. The properties associated with classes are also expanded and linked with the classes as shown in Figure 1<sup>1</sup>. This model is based on the expandable/collapsible trees that grow up according to the nodes that are being expanded.

It defines its own symbology for representing the diverse characteristics: for instance, Figure 1 highlights a green-coloured rectangle, the symbols representing the disjointWith and inverseOf characteristics. The main disadvantage of this modelling tool is that it represents all the information in the same visualisation, even duplicating nodes as depicted in Figure 1 with the

class AtomicProcess and the property realises; both cases are highlighted in blue. Duplicating elements result in a less efficient and redundant model and this redundancy causes the user to become easily confused when navigating through the model. The strategy of showing all the information in the same visualisation with redundancy makes it extremely difficult to navigate through a large ontology. Moreover, there is not a clear representation of the taxonomy, from the point of view of an isolated hierarchy of concepts to navigate through.

TopBraid Composer is an enterprise-class modelling environment for developing Semantic Web ontologies and building semantic applications. There are three available versions: a Free Edition, Standard Edition and Maestro Edition. TopBraid Composer is a UML-based modelling plug-in eclipse, part of the TopBraid Suite. TopBraid Composer is a fully Protégé-based tool that performs the most common operations over ontologies, such as: inference, consistency checking as well as the inclusion of SPARQL query engine. Figure 2<sup>2</sup> illustrates a view of the TopBraid Composer tool, composed by diverse visual components. The look and feel is similar to Protégé ontology editor, with the possibility of displaying a UML-like view. Even

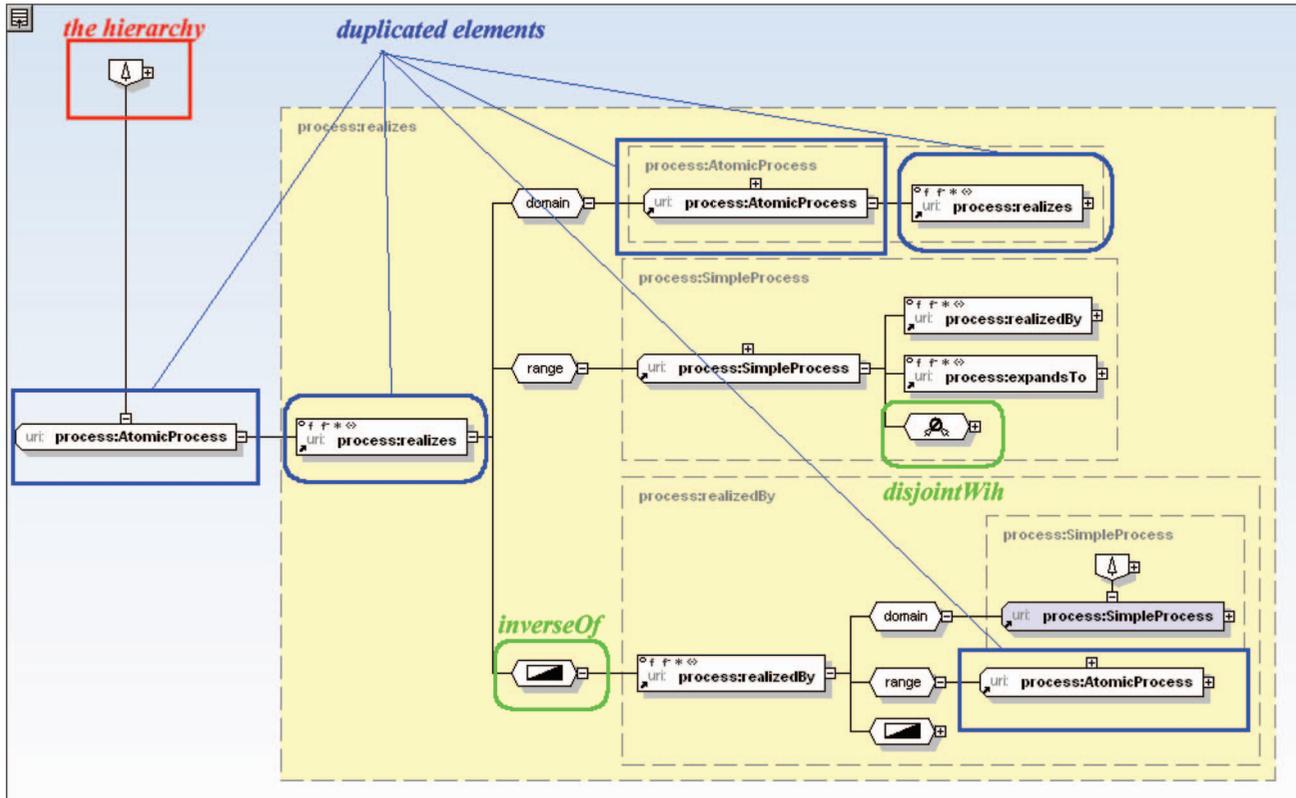


Figure 1. Visualising the SemanticWorks tool modeling environment. It is based on the use of conceptual maps for representing relations and classes in an ontology.

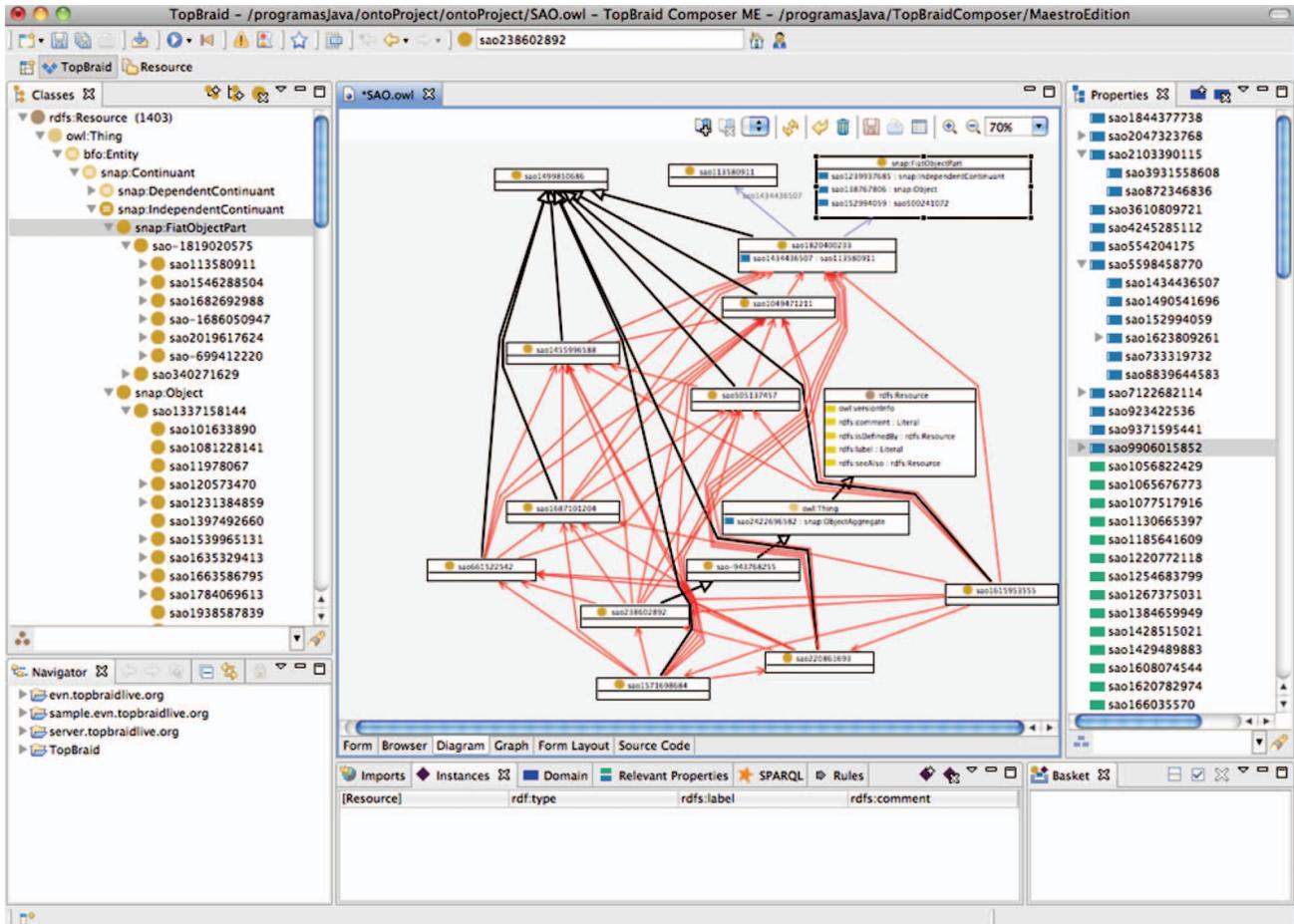


Figure 2. The TopBraid Composer modeling tool is based on the look-and-feel of Protégé. It uses a directed-graph visualisation, as well as UML class diagrams for modeling ontologies.

though it represents a good option, as the visual representation is based on the use of directed graphs and UML class diagrams, however this makes the visual analysis process difficult to perform for medium or large ontologies. It is based on the use of diverse panels containing diverse sources of information about the ontology.

IBM Integrated Ontology Development Toolkit (IODT) is a toolkit for ontology driven development, including EMF Ontology Definition Metamodel (EODM), EODM workbench, a Scalable Ontology Repository (SOR, before version 1.5 it is named Minerva) and some extenders to the core components. EMF Ontology Definition Metamodel is derived from the OMG Ontology Definition Metamodel (ODM) and implemented in the Eclipse Modeling Framework (<http://www.eclipse.org/modeling/emf/>). It includes RDFS/OWL parsing and serialisation, reasoning, and transformation between RDFS/OWL and other data-modelling languages. It is also an open source project of Eclipse.org (<http://www.eclipse.org/emft/>

projects/eodm/) that supports ontology building, management and visualisation. It has UML-like graphic notions to represent OWL classes, restrictions and properties in a visual way. It can also have multiple views to support visualisation of an ontology; these views are independent but synchronised, so changes made in one visualisation affect all of them. The Visual Workbench represents the model in a UML-like view, completely based on the IBM Rational for developing software. Advantages of this tool include the widely known UML standard representation of classes, properties, the hierarchy and the facility to create diagrams and the comprehensibility of them. The main disadvantage of this tool is the scalability to model a large ontology. As it has been described above, the TopBraid Composer also implements a UML-based visualisation. It seems that this project has not been updated for some years, apart from being a commercial product.

Protégé (<http://protege.stanford.edu/>) (Knublauch *et al.* 2004, Horridge *et al.* 2011) is a free, open source

ontology editor and knowledge-based framework. The Protégé platform supports two main ways of ontologies modelling: Protégé-Frames and Protégé-OWL editors. It also includes diverse plug-ins developed and maintained by the community. One of these plug-ins is called OWLViz, a graph-based visualisation that represent classes, properties, hierarchy and the classical tree of hierarchies view. Classes are represented as nodes in the graph, while properties are represented as edges connecting nodes, where the edges represent ‘is-a’ relations (hierarchy). Jambalaya (Lintern and Storey 2005, Storey *et al.* 2001) is another plug-in intended to visualise OWL ontologies with Protégé. The Jambalaya project can be found on its official site (<http://webhome.cs.uvic.ca/~chisel/projects/shrimp/demo/applets.html>). It is a complete plug-in that visually represents the components of the ontology and its relationships divided into two views, where each view can be displayed using one of six different layouts: grid, radial, spring, sugiyama, tree and treemap<sup>3</sup>. The view shown on Figure 3 is based on traditional graphs where relationships are represented in the same manner as in the self-contained model. This graph connects classes with classes (is-a relationships and object properties representing coupling relationships

among classes) as well as classes with their instances. Although Jambalaya represents a good tool to visualise an ontology, scalability is the main disadvantage due to large graph visualisations are well known for becoming cluttered, as shown in Figure 3<sup>4</sup>.

There are mainly two modelling approaches for ontologies: the first is based on the use of the well-known graph theory (Protégé, SemanticWorks) and the second approach is based on the use of UML class diagrams, such as in the Object Oriented approach (TopBraid Composer, IODT). Both approaches share almost the same problems; the first one is a lack of layout, especially for graphs and UML-based solutions. A layout refers to the organisation of the visual elements across the available visual space; it represents a crucial aspect in order so that the user can easily create a mental map and understand the represented information. For instance, Figure 2 depicts a Top Braid UML class diagram, where the edges coloured in black represent the ‘is-a’ relationships, defining the taxonomy of concepts. This diagram makes it difficult for the user to really understand the taxonomy of concepts. This is a crucial aspect, because a user looking at reusing an ontology has to perform a mental map of the taxonomy of concepts. For large or deep

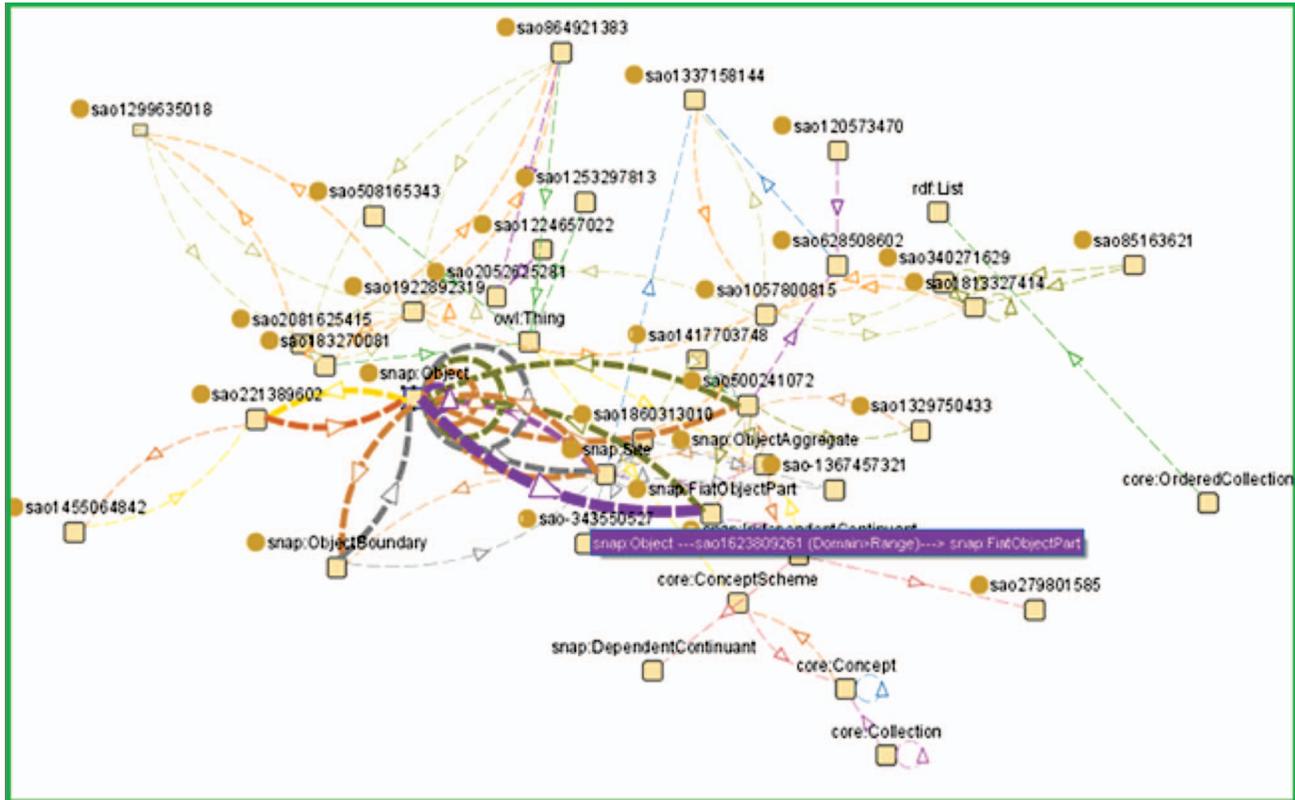


Figure 3. Jambalaya tool directed-graph visualisation, where relationships are represented as coloured arrows, and the arrowhead indicating the direction of the property, from the domain to the range.

ontologies, the use of the simple tree-like view such as Protégé, TopBraid Composer, IODT and most of the current tools, result in a less efficient way to navigate through the hierarchy of concepts. Figure 4 clearly depicts this situation: on the left side, the hierarchy of concepts and the right side a visualisation tree representing the taxonomy as well as the properties. The visualisation tree is almost useless, because of its layout it is not efficient, as most of the visual elements are overlapped, leaving diverse blanks around the screen. The results are very difficult for the user to understand and to see what is being represented there. Even when Jambalaya provides different layouts, most of them completely fail as can be shown in the reference (<http://www.analiticavisual.com/juan/caseStudy.html>).

A second problem with these visualisations is that they represent too much information; for instance, Figure 1 depicts a Semantic Works diagram, which grows according to the diverse element that is being

selected. The same diagram illustrates the hierarchy, properties, characteristics, etc. resulting in a big diagram full of different ‘roads’ to follow, making it difficult to create a mental map of the concepts.

To summarise, as it has been previously discussed, one of the main problem is the saturation of information in the visual models. Another important aspect resulting in unclear visual representations is related to the use of scroll bars. When a visualisation excessively grows, it needs to make use of scroll bars, causing the user to lose the context. This situation occurs with the Zooming functionality of Jambalaya and with Semantic Works. A third problem is the redundancy of information such as Semantic Works, and the saturation of information in the visual representations. Most of these problems are caused by the fact that these tools have been built putting together diverse widgets (panels, boxes, buttons, etc.), and the visualisation is in one sense a complement or plug-in to adapt. Their main structure is not based on

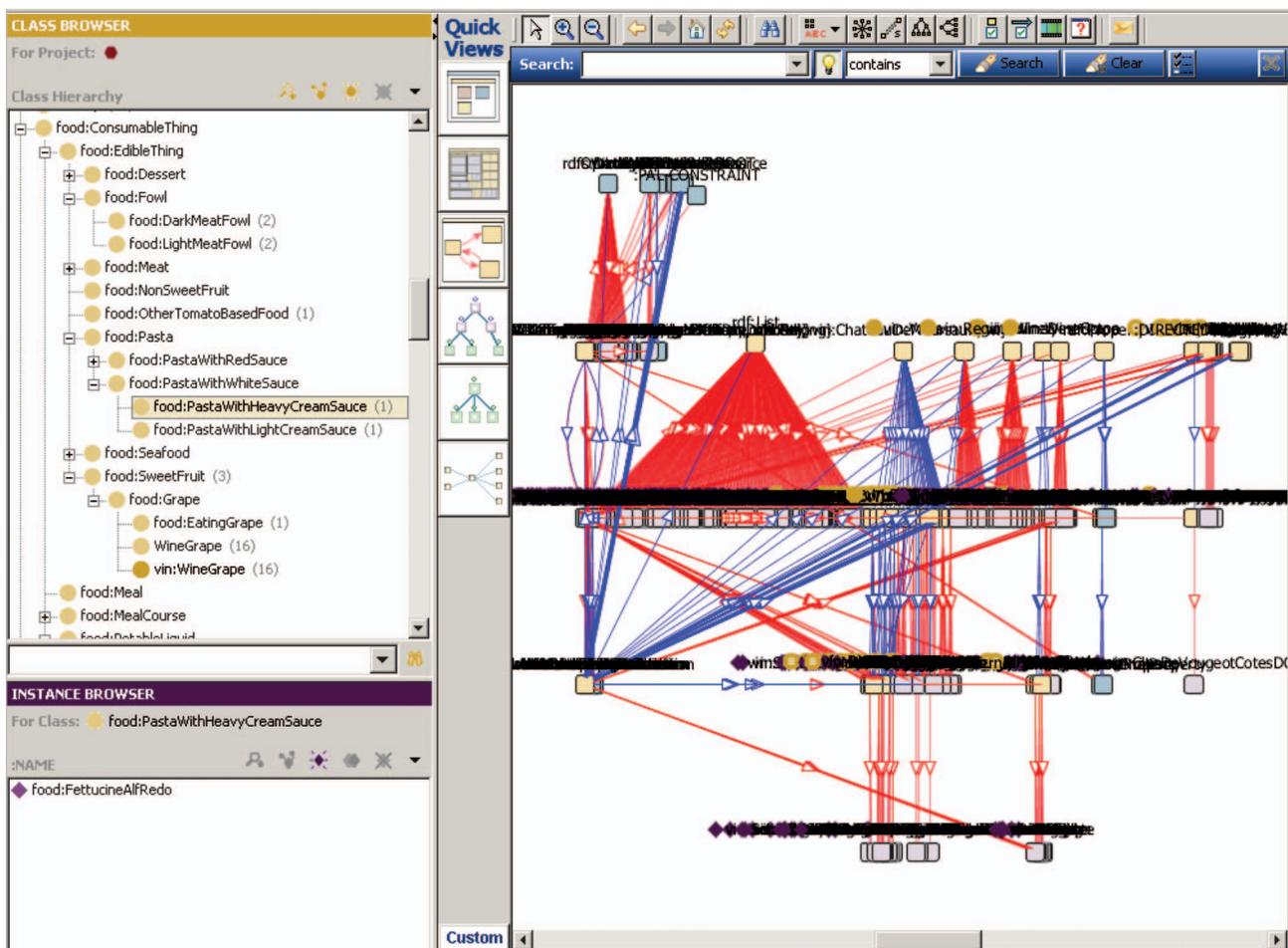


Figure 4. Showing the use of Protégé (Jambalaya) with the ontology Wine. On the left side, the hierarchy is shown as a tree-like view; while the right side shows a view using a tree layout with the hierarchy, properties and individuals.

the use of visualisations, but connecting diverse panels and displaying diverse information. They are basically based on the use of diverse panels with scroll-bars that grow-up according to the user selections, or the size of the information displayed, which results in the loss of the context.

From the point of view of an ontological engineer that follows a certain workflow, this is a less efficient way to organise the information. The specialist, or the user, is not able to perform diverse tasks at the same time, such as navigating the hierarchy, analysing the properties, analysing the individuals, etc. The strategy of displaying all this information makes it less efficient in the visual models, as will be contrasted to the OWL-VisMod approach. Table 1 summarises the most common problems detected with other tools and contrasts these problems with the solutions of OWL-VisMod. All these solutions have been evaluated by users, and the results are described in (García *et al.* 2011c). In addition, some of these visualisation techniques have also been successfully applied to diverse fields (Babaria 2004, Keim *et al.* 2008).

### 3. Analysing an ontology with OWL-VisMod

The creation of ontologies involves diverse phases, some methodologies and tools have been proposed (Corcho *et al.* 2003). One of these phases corresponds to the implementation of the ontological model in a specific ontological language, such as RDF, RDF-SCHEMA or OWL. Another phase included in most of the methodologies, is related to reusing ontologies. Reusing existing ontologies represents a practice that would be worthwhile from the point of view of ontological engineers, which would let them save time and effort. Diverse activities are involved in the reuse process, starting with a search of the ontology,

and ending with the process for the reuse of the ontology. Each one of these activities is introduced in detail in Figure 5.

Figure 5 illustrates the process of reusing an ontology which involves a phase of analysis of the existing ontology. The results of this phase are crucial, in order to take the correct decision about the value of reusing an ontology, and should be performed in the most efficient way possible. Here is where the Visual Analytics approach can be used, in order to perform analysis of information more efficiently to acquire knowledge. The reusability of ontologies plays an important role in the ontological engineering field and it is a very common task that can become complex without the adequate use of techniques for analysis. The phase of analysis involves the development of the main activities defined in the methodologies (Methodology, DOGMA, etc.), but instead of creating new elements (classes, properties, etc.), the analysis focuses on searching in the ontology for the semantic equivalency of these elements. Previous and after the phase of analysis, the workflow follows a visual analytics approach for discovering knowledge, based on the use of visualisations and interaction.

The visual representations and interaction technologies provide the mechanism for allowing the user to see and understand large volumes of information at once. The human mind can understand complex information received through visual channels. Visual analytics builds upon this ability to facilitate the analytical reasoning process (Thomas and Cook 2005). OWL-VisMod has been conceived to implement diverse techniques taken from the visual analytics (Keim *et al.* 2006) and Information Visualisation (InfoVis) (Chen 2006, Kerren *et al.* 2008) fields, in order to enrich the conceptual modelling process of a knowledge base.

Table 1. It provides a summary of the most common problems detected with the diverse analysed tools and contrasts these problems with the solutions implemented in OWL-VisMod.

| Common problem                            | Improvements with OWL-VisMod   |
|---|--|
| Lack of layout                            | All the visualisations in OWL-VisMod are provided with a specific layout, such as: self-contained layout (Treemap), structure layout (Tree), radial layout (Semantic Zooms, Global Coupling)   |
| Saturation of information                 | OWL-VisMod has defined a workflow starting from the most general aspects to the most specific. The information is being displayed according to the user interaction, making use of blurring (Focus+Context) to blur the context as background, avoiding the saturation of visual elements  |
| Redundancy of elements                    | OWL-VisMod avoids the redundancy of visual elements; it represents a unique element in each visualisation  |
| Poor visual expressiveness                | OWL-VisMod is based on the use of intuitive visual elements, such as figures (ellipses or rectangles), distinctive colors (for properties, individuals, edges) and symbols (clusters of elements, edges)   |
| Lack of workflow and saturation of panels | OWL-VisMod has defined a specific workflow starting from the most general aspects to the most specific. The whole space available in the screen is provided to a visualisation, in order to saturate the visual context with panels full of functionalities. Moreover, OWL-VisMod does not implement scrolling bars, because they cause the context lost |

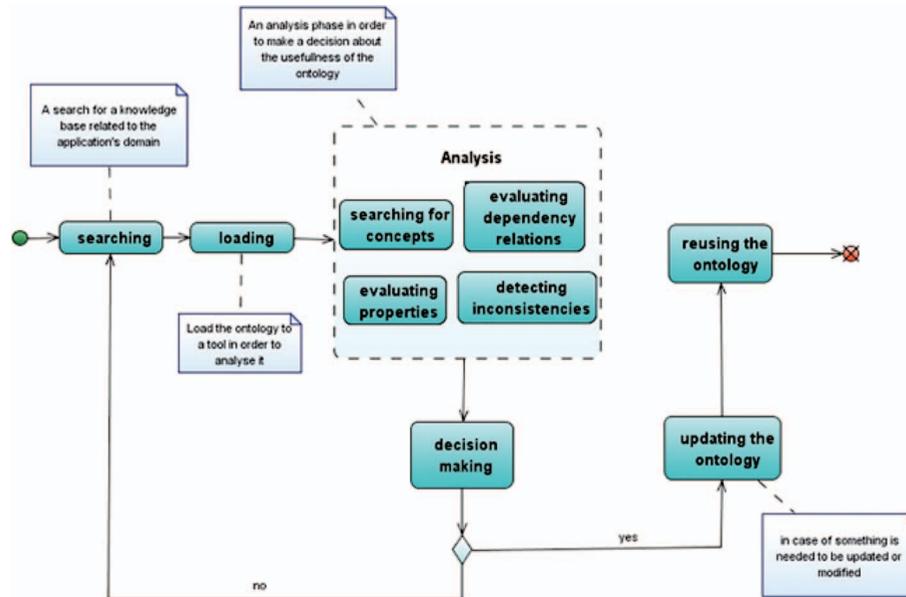


Figure 5. Diagram that describes the main activities during the process of the reuse of an ontology. Highlighting the phase of analysis, crucial in making the decision of whether or not to reuse an ontology.

OWL-VisMod implements a Visual Analytics process (Keim *et al.* 2010) characterised through the interaction between data and visualisations in order to discover knowledge. The interaction represents an essential aspect in the process of knowledge discovery, and it lets the user modify the views, to query the models, to go into the details of any element or request different perspectives of the data. Figure 6 illustrates the visual analytics process defined in Keim *et al.* (2010), where the different phases of the knowledge acquisition process are depicted. OWL-VisMod follows an equivalent process, completely based on the visual analytics approach, where the data is an OWL ontology; then, a conceptual model is built and visualised, letting the user interact with it, and to finally acquiring the knowledge represented in the ontology. The main and final target is to acquire this knowledge in order to make decisions about it, such as the reuse of the ontology or not and if so then how it needs to be modified, etc.

OWL-VisMod can be used to support: loading OWL ontologies, analysis of their schema (classes, properties and individuals), conceptual modelling of OWL ontologies (generation of OWL-DL and OWL-Lite code from a conceptual model), definition of rules distinguishing concepts (constraints, restrictions and characteristics), calculation of diverse metrics (schema and coupling) and the development of the lifecycle of an ontological model. On the other hand, OWL-VisMod is not intended to encompass the specification of the proof theory or inference rules, free and

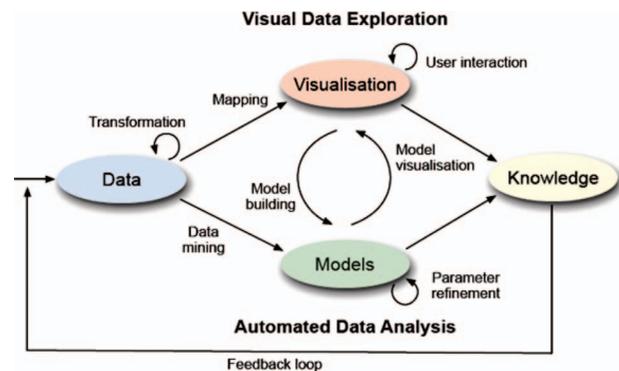


Figure 6. Illustrates the different phases of the visual analytics process. An analogous process is followed by OWL-VisMod to discover knowledge.

conditional logics, matching and alignment of ontologies.

To illustrate this proposal, the case study follows a workflow described in Figure 5; which starts with a global view of the taxonomy, so the user can navigate and search for concepts and clearly understand the whole hierarchy (superclasses). In the case study, the user is interested in a specific concept, by just selecting them the details are displayed, but it is important to leave the context, as this aspect lets the user know exactly where he or she is located, as well as where is he or she going to. This approach improves the other existing tools, because they just put together diverse panels with functionalities, but do not define a clear

workflow, nor do they use the available screen space in an efficient way. The advantage of using visual elements is that the life cycle of ontologies is easier to develop as well as being more intuitive resulting in a more effective process, compared to the rest of the previously discussed tools that are not efficient in dealing with visual interactions. The Visual Expressiveness—a concept related to the quality of the visual representations—is poor, resulting in visualisations that become completely useless because they are unclear, difficult to navigate, complex, redundant and overcrowded, as shown in Figures 1–4. These tools are poor visually speaking, and instead of helping in the process of abstraction of a certain knowledge base, they become so complicated that users cannot use them, and they become completely useless.

OWL-VisMod has been developed in Java and represents an upper modelling tier; it uses Jena API (<http://jena.sourceforge.net/>) to manage OWL ontologies. The Jena framework is responsible for loading and managing ontologies. OWL-VisMod queries directly over the Jena framework, and builds some data structures to manage the ontology. This information is processed, managed by the framework and finally visualised. The tool generates the OWL code representing the ontological model, and it is also provided with File Transfer Protocol (FTP) capabilities so that the generated code can be published online. It is based on the use of various visualisations and linked views that let the user to perform analysis of all the aspects involved in the ontology modelling process.

In order to illustrate the use of OWL-VisMod in a real scenario, a case study is described in the following sections, with some videos and images (with a higher resolution) that are available online. These videos and images help to support and explain the process. This case study is related to the development of the project called Turimov, a Mobile Support Intelligent System for supporting the tourism in the region of Castile and Lion, Spain. The main objective of this project is to develop a system based on the use of diverse mobile technological platforms, to support diverse activities that help tourists to search for places, events, activities, etc. during their trip across the Castile and Lion region. More specific details can be found on the Interra company website (<http://www.interra.es>).

The Turimov project is based on the use of OWL ontologies to manage all the available information such as places, events, or activities, as well as profiles for users, which suggests specific trips to the user according to their interests. All the information is stored using ontologies, and they play a crucial role in the process of performing inference over the data, to create trips and diverse suggestions according to the user's profiles. The case study for the project Turimov

follows the workflow introduced in the diagram shown in Figure 5, starting with a search of an ontology over the internet. Once a candidate ontology has been found, the phases of loading, analysis and decision making are performed in order to reuse it. The following subsections describe each of the phases in the analysis process of the ontology, starting with the hierarchy (to search for some concepts), followed by the analysis of properties and individuals and the global coupling of the classes. The last subsection, describes the decision-making process, as well as the reasons for deciding on the reusability of the ontology.

### 3.1. *Analysing taxonomies of concepts*

Diverse searches were carried out, having found one Ontology called wine.owl (<http://krono.act.uji.es/Links/ontologies/wine.owl#>) that could be useful and be reused and included in the Turimov project. This ontology has been defined by the TKBG (Temporal Knowledge Bases Group) research group from the University Jaume I in Castellón Spain, perhaps to be used in a certain type of research. The information provided indicates that this ontology has been derived from the original DAML (DARPA Agent Markup Language) Wine Ontology<sup>5</sup>. In order to decide whether or not it can be reused, a complete analysis has been done on it to identify the changes that should be made.

This process started by loading the ontology from its URL<sup>6</sup>. Once the ontology is loaded in OWL-VisMod, the analysis phase starts by navigating its taxonomy in order to evaluate the defined concepts, to detect the ones that could be used in the new project, as well as those new ones that should be added to this taxonomy. The ontology contains diverse regions where wine is produced; it also defines diverse types of wines as well as other characteristics. Another aspect that is included in this ontology refers to meals and their varieties. A video illustrating this analysis is available<sup>7</sup>, to support and explain the workflow, so that there is a better understanding of the process. The analysis requires the search of two specific aspects: information related to Spanish wines and Spanish cuisine. The project is targeted on a specific region in Spain, so the information should be focused on both these aspects.

According to the analysis<sup>8</sup>, it has been detected that this ontology does not include concepts from Spain; authors considered other regions, such as Italy, France, USA, etc. but they have not considered the region of Spain. Figure 7 depicts the taxonomy of the ontology, showing the diverse concepts defined in it. As can be clearly seen from the image, this taxonomy does not include the region of Spain, nor does it include Spanish wine or cuisine.

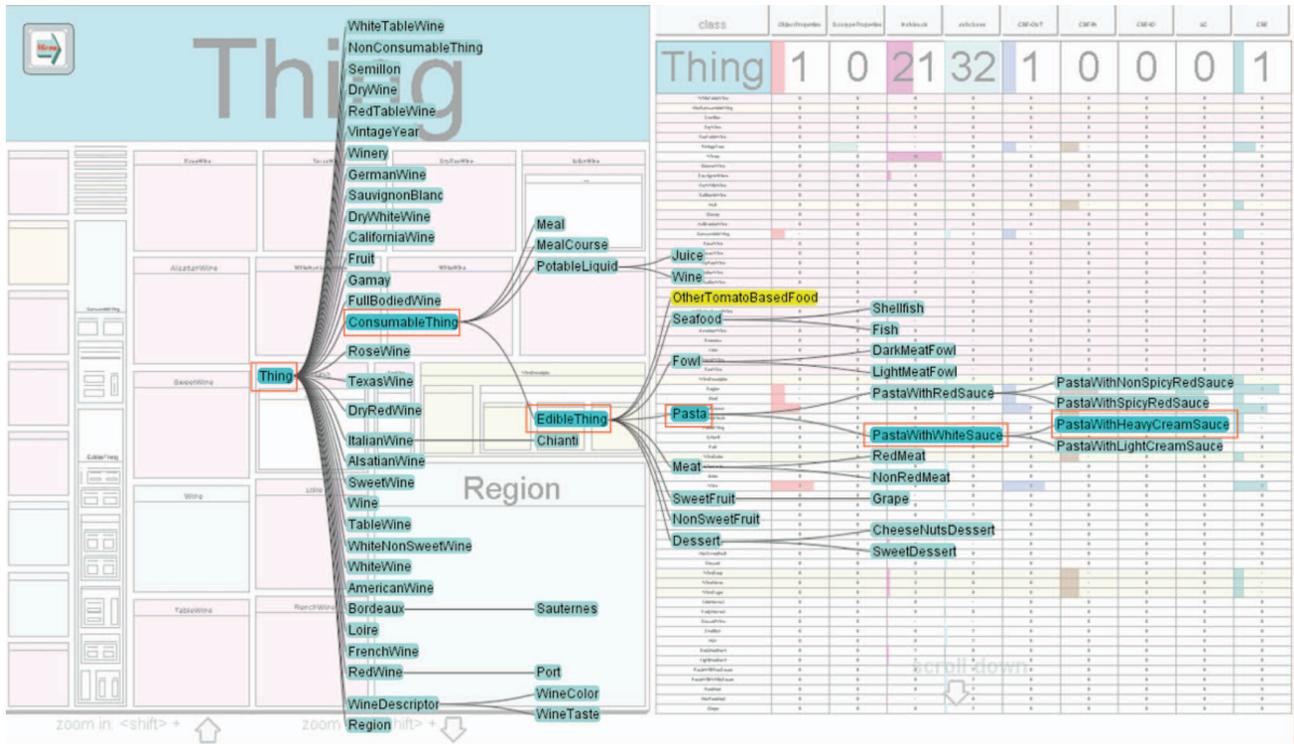


Figure 7. The tree represents the taxonomy of the Wine Ontology. After analysing all the concepts defined by the authors, the region of Spain as well Spanish wines and cuisine were not included.

Figure 7<sup>9</sup> illustrates the use of the tree visualisation to display the taxonomy, showing the trace from the root node to a certain selected concept. As a contrast to the representation of hierarchies in other tools such as Semantic Works<sup>10</sup>, Top Braid Composer<sup>11</sup> or Jambalaya<sup>12</sup>, the results of this technique are more intuitive. This can clearly be seen in Figure 4, as the left side represents exactly the same class as the tree (PastaWithHeavyCreamSauce). Owl-VisMod provides a more intuitive, clearer to understand and better visualisation of the taxonomy. Apart from the tree illustrated in Figure 7, a treemap (Johnson and Shneiderman 1991) visualisation is also used to represent hierarchies, both these techniques have been tested to be able to efficiently display large ontologies, and as well as they are intuitive, scalable and easy to navigate. The interaction with these visualisations is illustrated in the diverse videos available online.

Another important aspect to comment on is that most of the visualisations in OWL-VisMod implement the Focus+Context (Kosara *et al.* 2002), a visual technique based on providing a focus over a selected element, without losing the whole context. Leaving the context lets the users create a mental map of the whole visualisation, in contrast to the use of the physical zoom used by Jambalaya, which performs a zoom on a

specified element, but loses the rest of the context. Figure 7 displays a hierarchical tree, leaving the original treemap view of the ontology in the background as a context.

### 3.2. Analysing properties and individuals

Once the hierarchy has been navigated and in order to see the taxonomy of concepts, the next step involves going into the details of the classes in the ontology, in order to answer more specific questions, such as: what are the characteristics of wines (colour, flavour, sugar taste or the body)?, which types of wines are included in the ontology?, or from which regions and wineries do the wines come from? These questions can be answered by going into the details of the classes. The analysis<sup>13</sup> performed over the ontology indicates that, as it has been previously mentioned, the region of Spain has not been included, and neither have Spanish wineries. The types of grapes, as well as the wineries refer to in the majority of cases of French wines. On the other hand, general characteristics of wines<sup>14</sup> such as colour, flavour, sugar levels or the body have been included; these characteristics are valid for all wines, and they are not related to a specific region, which implies that these characteristics also allow us to classify the Spanish wines. This analysis has been

performed using a semantic zoom technique; a non-graphical zoom that lets the user see the different amounts of detail in a view, it involves changing the type and meaning of the displayed information. In contrast to a graphical zoom that only changes the size of the selected element, semantic zoom lets the user know something else that had been previously hidden.

Figure 8<sup>15</sup> depicts two different views of the semantic zoom technique for two classes. This technique is achieved by dragging the class out of the treemap which you want to view in detail. Figure 8a shows all the different kinds of grapes, included in the Ontology, using a radial graph layout (Keim *et al.* 2006), where the type CabernetSauvignonGrape is highlighted. The visualisation is defined by a typical UML-like class representation, having three buttons for selection: object properties, datatype properties and individuals. Figure 8b depicts the object properties of the class Wine (madeFromGrape, hasWineDescriptor, hasColor), as well as a cluster showing the equivalent classes. In this case, the same class Wine is equivalent with itself and in a clear case of an inconsistency, or a redundancy and useless definition; there is no coherent reason for defining a class equivalent with itself.

3.3. Analysing the global coupling of classes

OWL is a rich vocabulary description language for describing properties and classes, such as relations between classes, cardinality, equality, rich typing of

properties, characteristics of properties and enumerated classes. OWL ontologies distinguish between two main categories of properties: object properties and datatype properties. Object properties relate classes in the domain with classes in the range while datatype properties relate classes in the domain with simple data values in the range. Coupling among classes is given by the object properties and these coupling relationships define the way that classes interact with each other.

OWL-VisMod has implemented a specific visualisation focused on representing the coupling among all the classes in the Ontology, based on the Holten’s approach proposed in Holten (2006), for software engineering projects. It basically organises all the classes in a radial layout; the classes are collocated according to their CBE (cost-based evaluation) metric value, calculated as it has been defined in (García *et al.* 2010a). By organising the classes according to their CBE metric value, the most coupled classes will be located at the top of the visualisation while the less coupled ones are organised in the clockwise sense from the circumference. This is done in this way in order to easily detect the most coupled classes that usually are the most important classes in the ontology. The coupling is represented using an edge that connects both coupled classes; this edge is defined by a Bezier curve and directed by the control points nearest to each of both classes. Two or more classes are said to be coupled when they are related by a property; this property defines a specific conceptual relationship among them. This approach also defined in Holten

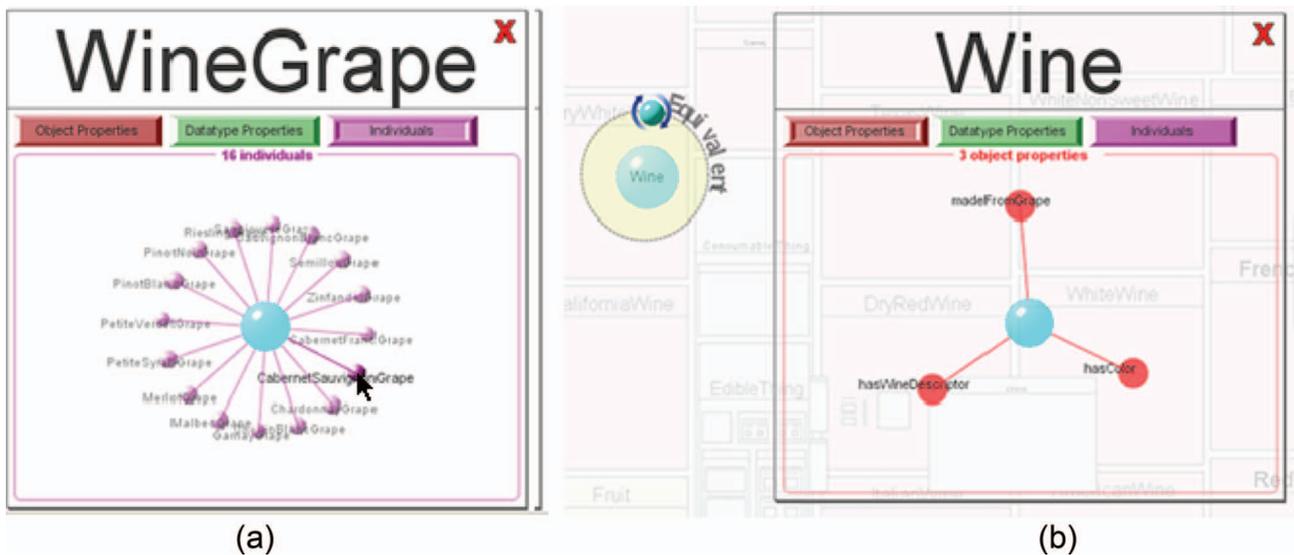


Figure 8. The Semantic Zoom technique shows the internal details of a class. Figure (a) depicts the individual for the class WineGrape, corresponding to all the types of grapes in this classification. Figure (b) shows the object properties for the class Wine (madeFromGrape, hasWineDescriptor, hasColor), as well as a cluster showing the equivalent classes.

(2006) is intended to simulate forces that strength the edges in order to get a clearer visualisation.

Figure 9<sup>16</sup> depicts the global coupling of the wine ontology. As it is shown, this ontology is less coupled, having just 17 properties, 16 of them being object properties and just 1 of them a datatype property. This means that only 16 of the 81 total classes in the Ontology are coupled, i.e. less than 20% of the classes. The class Region is the most coupled class in the Ontology, followed by the classes MealCourse and Wine. This analysis<sup>17</sup> is focused on the relations among diverse classes, in order to see which concepts are related, as well as the relations that should be added for the Turimov project.

In this ontology, the class Wine is the main concept from which other concepts are included. Figure 8b shows the object properties defined in the class Wine, nevertheless, the classes related to it are not specified, and it is not the purpose of this visualisation. In

contrast to this, Figure 9 highlights the names of the coupled classes in the list at the right side of the visualisation. Moreover, the relations and the spheres representing these classes are also highlighted, those being WineGrape, WineDescriptor and WineColor. The variation of tonality in the edges indicates that the class Wine belongs to the domain in these three properties. This situation will be explained in detail in Figure 10a<sup>18</sup>.

By selecting the class, a semantic zoom is performed and the coupling visualisation shown in Figure 10a is displayed central to the visualisation. The properties are defined by having a specific domain and range of classes; the domain of a property limits the individuals to which property can be applied, while the range of a property limits the individuals to what the property may have as its value. The direction of a property is generally taken from left to right to clearly indicate the classes belonging to the domain and the

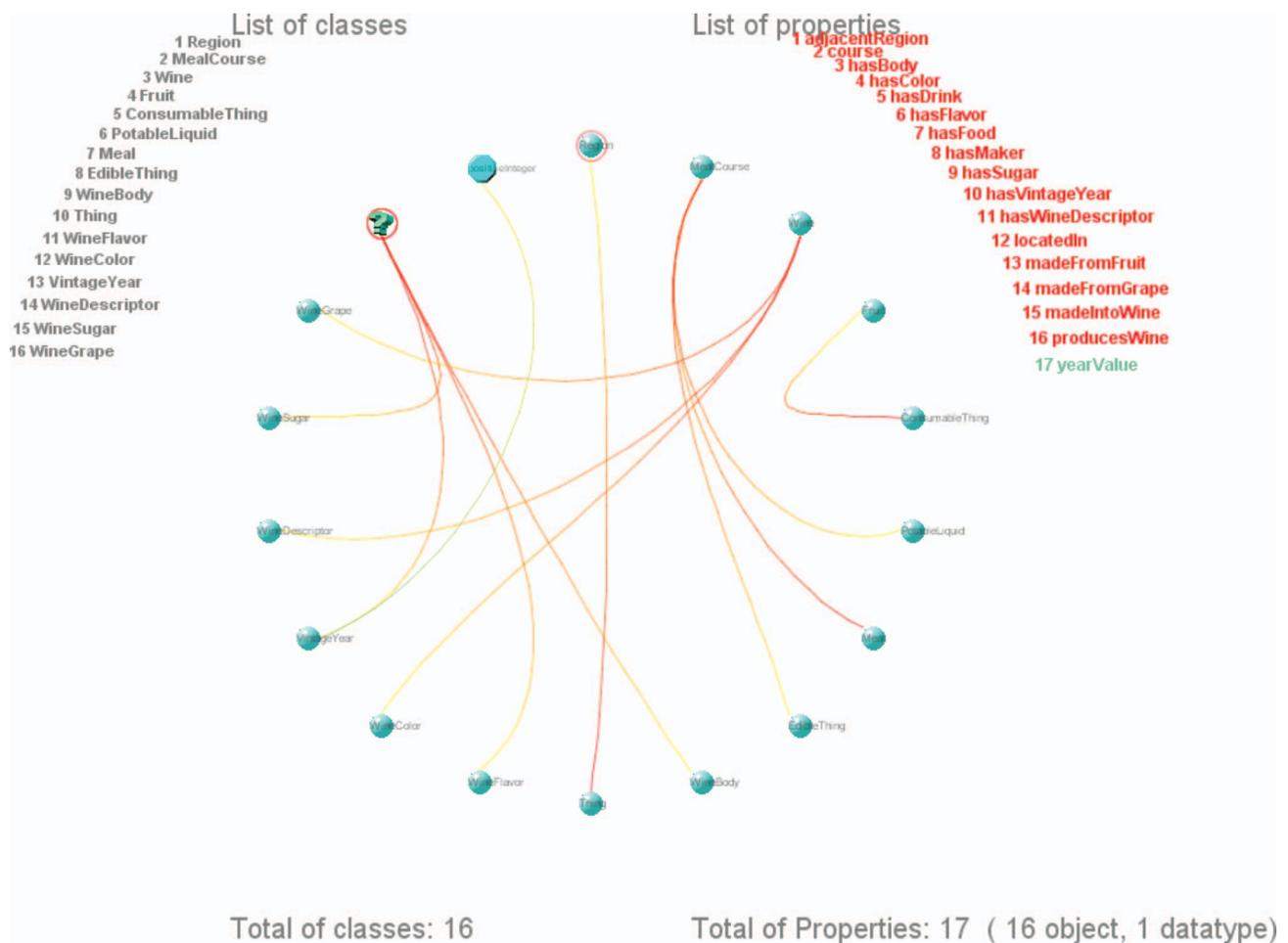


Figure 9. The Semantic Zoom technique shows the internal details of a class. Figure (a) depicts the individual for the class WineGrape, corresponding to all the types of grapes in this classification. Figure (b) shows the object properties for the class Wine (madeFromGrape, hasWineDescriptor, hasColor), as well as a cluster showing the equivalent classes.

classes belonging to the range. This visualisation has been proposed and fully-explained in detail in García *et al.* (2010b) as well as diverse coupling metrics (García *et al.* 2010a). The semantic zoom shown in Figure 10a and b also includes the inherited properties by the class, from all its superclasses in the taxonomy; these properties are indicated with a dashed contour line in the visualisation.

Figure 10b depicts the class Region, the most coupled class in the ontology. This class is intended to provide a description of a specific region where the wine comes from. It is provided with 36 individuals, containing different places from different countries, where wine is produced. An inconsistency of the wine ontology is that some relations have not defined the domain or the range. These properties are also included in the class Region, as in consequence of this, an inference process should be done in order to decide the class in the domain or the range.

One of the relations highlighted in Figure 10a is *madeFromGrape*, having the class *Wine* in its domain, while the class *WineGrape* belongs to the range. These two classes were separately analysed in another context in order to know their internal values as shown in Figure 8a and 8b. Now the analysis is focused on a different perspective, in order to find out the relationships that link both these concepts, instead of the details of their internal components. In this context, both concepts are seen as two entities that share a property that injects a dependency between them. As a mental abstraction, this dependency would be logically interpreted as saying that ‘a Wine is made from

WineGrape’, where the concept *WineGrape* is interpreted as a kind of grape to be used in the production of wine. This dependency is also represented in Figure 11<sup>19</sup> but from the point of view of the property itself. This knowledge is important because it means that this ontology is only considering wines from the most common context, wines that are made of grapes, while there are other kinds of wines made from other fruits or plants that are not being considered in this knowledge base, such as the gooseberry wine, currant wine, etc.

The coupling as a dependency relationship between two concepts, in this case *Wine* and *WineGrape*, is visualised in Figures 9–11, from three different perspectives. Figure 9 illustrates a general view of the global coupling for all the classes, but highlighting the class *Wine*. On the other hand, Figure 10a represents a semantic zoom visualisation of the coupling of the class *Wine*, being the proper class at the centre of the whole visualisation, as the focused element, while the rest of the elements surrounding it represent the related concepts through the properties. Finally, Figure 11 also depicts this knowledge, but from the point of view of the property, meaning that now the focused element is the property, and the concepts belonging to the domain or the range of it are grouped in clusters of classes; in this case there is just one class in either domain and range, but there could be more than one class.

Another aspect to highlight is the fact that the coupling among classes is poor, resulting in disconnected classes and a lack of information. For instance,

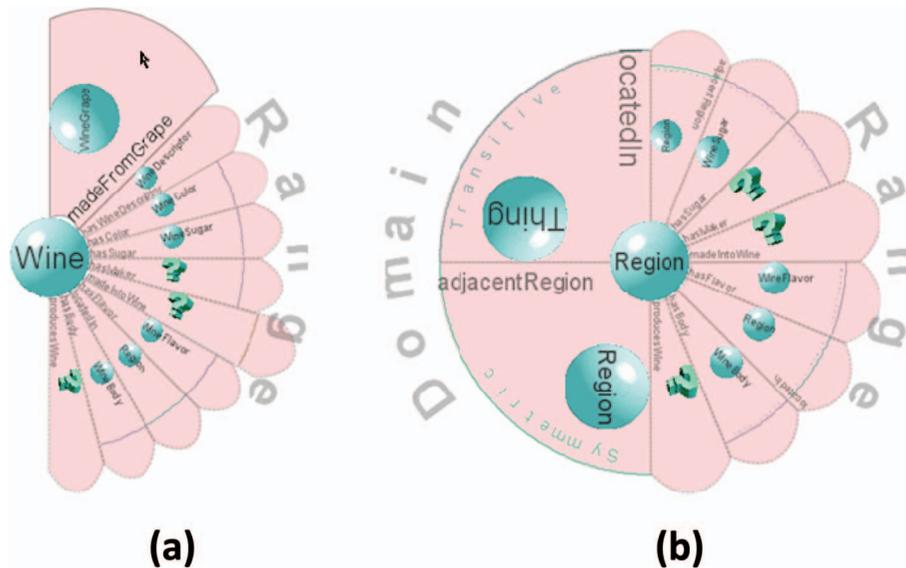


Figure 10. Figure (a) shows the semantic zoom of the class *Wine*, with all its relations and coupled classes. Figure (b) depicts the semantic zoom over the most coupled class *Region*. The lack of specific classes is highlighted using a question mark, helping the user to identify such inconsistencies.

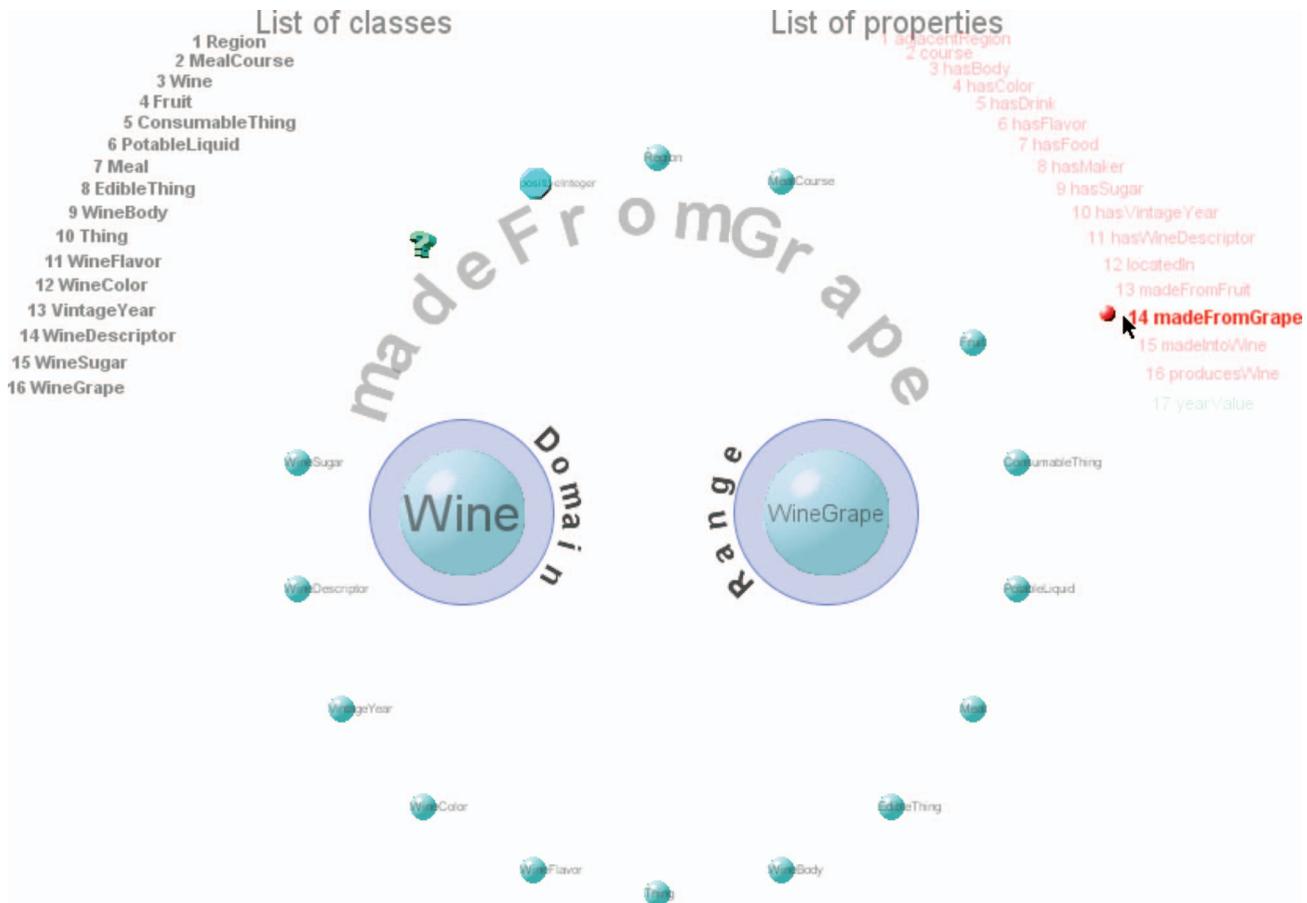


Figure 11. A view of the property `madeFromGrape`, showing the classes in the domain (`Wine`) and the classes in the range (`WineGrape`).

the class `Winery` has 43 instances corresponding to the same number of wineries around the diverse regions included in the ontology. The problem is that these instances cannot be referenced to a certain region, because there is no property relating to the classes `Region` and `Winery`. The class `Region` has 36 instances, which should be related to the wineries in order to define the region where a specific winery is located.

### 3.4. Decision-making process

The main objective of the analysis is to decide whether or not this ontology can be reused for the project `Turimov`, according to its specifications. The most important result of this analysis is the fact that neither the region of Spain nor Spanish wines have been included in the ontology. As the project `Turimov` is completely focused on and targets a certain region of Spain, the results of this aspect are crucial. This ontology defines the basic concepts about wines and some others such as food. Having this work done, a solution could be to populate the ontology with

instances that have specific values from the Spanish region. Moreover, it has seen that a small set of properties have been defined, so it should be enriched with more properties and perhaps some other concepts too. On the other hand, some inconsistencies have been also detected, such as some missed values for domain and ranges of diverse properties, and the evident redundancy of declaring the class `Wine` equivalent of itself. These inconsistencies can be solved easily, so they do not really affect the use of the ontology.

The decision-making process is supported by the analysis described above. Based on this, it has been decided to use this ontology with certain modifications, especially to make it adequate in order to meet the goals of the project. It is worth-while to reuse this knowledge base, in order to avoid the construction of a new one from scratch, as it is deemed to be time consuming and a labour intensive task. Sometimes, for reusing large ontologies, a modularisation process is required; in this case, the wine ontology is not very large, and it contains the basic concepts, so there is

no need for developing a modularisation process, but reusing it as provided.

The first step is to define the class SpanishWine, as a subclass of the class Thing, having the URL <http://data.turisem.info#><sup>20</sup>. Following the same schema as that of the Ontology wine, in order to be consistent with it, the class Spanish-Wine has been defined as an intersection with the class Wine. Another way to define it would be by declaring it as subclass of class Wine instead of as intersection. Once this class has been defined and included in the model, the second step is to define the different values for the age of the Spanish wine. In case of Spanish wines, five values are defined to classify the wines, according to their age. These values are: *Joven*, *Joven con Crianza*, *Crianza*, *Reserva* and *Gran Reserva*. An enumeration called AgingOf-SpanishWines including these values has been defined<sup>21</sup>. These values can be different for wines from other regions of the world.

The next step consists of the addition to the ontological model, of the Origin Denomination of the Spanish wines. The origin denomination refers to a geographical indication applied to a specific product, whose quality and characteristics depend on its geographical origin, such as the grapes in the case of wines. Within the region of Castile and Lion, there are five origin denominations: *Bierzo*, *Cigales*, *Ribera del Duero*, *Rueda* and *Toro*. These values have been included as an enumeration of values<sup>22</sup>, called OriginDenomination, and subclass of Region. Finally, the diverse internal regions of the Castile and Lion region itself are also added to the knowledge base. These nine regions are: *Avila*, *Burgos*, *León*, *Palencia*, *Salamanca*, *Segovia*, *Soria*, *Valladolid* and *Zamora*. The enumeration called CastillaYLeon has been defined as a subclass of Region<sup>23</sup>.

Other modifications to the ontology can be performed; nevertheless, the previously described results are crucial in order to see if the ontology can be reused for the project Turimov. Perhaps, during the development of the project, new concepts, relations or information should be added to this ontology, in order to make it adequate for its use in this tourism project.

#### 4. Conclusions

The development of ontologies represents a crucial aspect in the Knowledge Engineering field. In this paper, OWL-VisMod has been used to analyse an ontology in order to provide a means of assessment and evaluation in order to support the decision-making process on whether or not to reuse it. This paper has proposed a workflow for the OWL analysis process, which is based on two ontology building methodologies: Methontology and DOGMA. This workflow defines

each of the phases targeted to study the viability of reusing an existing Ontology, from the perspective of the semantic binding, which refers to the identification of concepts that are semantically equivalent, and can be used in diverse domain applications.

In addition, the visual analytics techniques also support the development of the workflow, letting the user perform an analysis in order to acquire knowledge and to make decisions. This analysis has led to: firstly, understand the ontology, by navigating over its taxonomy of classes, relations and attributes; secondly, to detect some inconsistencies; and thirdly, which is the main aspect, gain the insight to decide whether this ontology can be reused for the new project. Besides, it also enables to clearly identify which new classes, attributes or relationships that need to be added, to the new requirements and specifications.

OWL-VisMod integrates diverse techniques that have been widely and successfully used in diverse application fields. This approach offers the possibility to perform a visual analysis of the information, from diverse perspectives, enriching the abstraction process, and the knowledge acquisition. The interaction represents an essential aspect in the process of knowledge discovery. OWL-VisMod supports an interaction process, letting the user perform diverse activities to analyse the knowledge base by means of different views that enrich the analysis process.

OWL-VisMod improves the conceptual modelling process in contrast to the tools described in Section 2, which are less efficient (as shown in the diverse figures above). Moreover, the main disadvantages of these tools, such as the lack of a layout, the overcrowding of visual elements, the redundancy of visual elements, and the use of scroll bars with huge visualisations that make the user loses the context and the difficulty of finding inconsistencies in the ontology, have also been pointed out. It is important to highlight that this analysis is exclusively from the point of view of the visual expressiveness of the tools, and is not intended to evaluate their functionality and usability.

#### Acknowledgements

This work was supported by Spanish Government project TIN2010-21695-C02-01, the Castile and Lion Regional Government through GR47 and is also supported by the Ministerio de Ciencia e Innovación of Spain under project FI2010-16234.

#### Notes

1. Figure 1 with a higher resolution as well as other related images can be found at: <http://www.analiticavisual.com/juan/semanticWorks.html>.
2. Figure 2 with a higher resolution as well as other related images can be found at <http://www.analiticavisual.com/juan/topBraidComposer.html>.

3. Diverse images using these layouts are available at: <http://www.analiticavisual.com/juan/caseStudy.html>
4. Figure 3 with a better resolution as well as diverse jambalaya related images are available at <http://www.analiticavisual.com/juan/jambalaya.html>.
5. <http://ontolingua.stanford.edu/doc/chimaera/ontologies/wines.daml>.
6. Illustrated in this video: <http://www.analiticavisual.com/juan/videos/video1.avi>.
7. <http://www.analiticavisual.com/juan/videos/video2.avi>.
8. Shown in the video: <http://www.analiticavisual.com/juan/videos/video3.avi>.
9. It can be seen online in a higher resolution at: <http://www.analiticavisual.com/juan/treemap.html>.
10. <http://www.analiticavisual.com/juan/semanticWorks.html>.
11. <http://www.analiticavisual.com/juan/topBraidComposer.html>.
12. <http://www.analiticavisual.com/juan/jambalaya.html>.
13. Shown in the video: <http://www.analiticavisual.com/juan/videos/video4.avi>.
14. Shown in the video: <http://www.analiticavisual.com/juan/videos/video5.avi>.
15. It can be seen online in a higher resolution at: <http://www.analiticavisual.com/juan/semanticZoom.html>.
16. Available with a higher resolution at: <http://www.analiticavisual.com/juan/coupling.html>.
17. Shown in the video: <http://www.analiticavisual.com/juan/videos/video6.avi>.
18. Available with a higher resolution at: <http://www.analiticavisual.com/juan/coupling.html>.
19. Available with a higher resolution at: <http://www.analiticavisual.com/juan/coupling.html>.
20. This process is illustrated in the video: <http://www.analiticavisual.com/juan/videos/video7.avi>.
21. This process is illustrated in the video: <http://www.analiticavisual.com/juan/videos/video8.avi>.
22. Illustrated in the video: <http://www.analiticavisual.com/juan/videos/video9.avi>.
23. Shown in the video: <http://www.analiticavisual.com/juan/videos/video10.avi>.

## References

- Babaria, K., 2004. *Using treemaps to visualize gene ontologies* [online]. Maryland: Human Computer Interaction Lab and Institute for Systems Research, University of Maryland, College Park. Available from: [www.cs.umd.edu/hcil/treemap/GeneOntologyTreemap.pdf](http://www.cs.umd.edu/hcil/treemap/GeneOntologyTreemap.pdf)
- Bera, P., 2007. *Using ontologies in the context of knowledge management systems*. Thesis (PhD). The University of British Columbia (Canada).
- Biggerstaff, T. and Perlis, A., 1989. *Software reusability: concepts and models*. New York: ACM Press.
- Biris, V., 2003. *Ontology and reuse in model synthesis*. Thesis (PhD). School of Informatics, University of Edinburgh.
- Chen, C., 2006. *Information visualization: beyond the horizon*. Berlin: Springer-Verlag.
- Colomo-Palacios, R., et al., 2010. A case analysis of semantic technologies for R&D intermediation information management. *International Journal of Information Management*, 30 (5), 465–469.
- Corcho, O., Fernandez-Lopez, M., and Gomez-Perez, A., 2003. Methodologies, tools and languages for building ontologies. Where is their meeting point? *Data and Knowledge Engineering*, 46 (1), 41–64.
- Davis, R., Shrobe, H., and Szolovits, P., 1993. What is a knowledge representation? *American Association for Artificial Intelligence* 14, 17–33.
- Fernández, M., Gómez, A., and Juristo, N., 1997. Methontology: from ontological art towards ontological engineering. In: *AAAI-97: Proceedings of the 14th national conference on artificial intelligence*, 27–31 July, Providence, Rhode Island. Palo Alto, CA: AAAI Press, 33–40.
- García, J., García-Peñalvo, F., and Therón, R., 2010a. Defining coupling metrics among classes in an OWL ontology. In: *Trends in applied intelligent systems. In: 23rd international conference on industrial engineering and other applications of applied intelligent systems, IEA/AIE 2010*, 1–4 June, Córdoba, Spain. Berlin: Springer. LNAI 6097, 12–17.
- García, J., García-Peñalvo, F., and Therón, R., 2010b. Visualising semantic coupling among entities in an OWL ontology. In: *Ontology, Conceptualization and Epistemology for Information Systems, Software Engineering and Service Science. 4th International Workshop, ONTOSE 2010*, 7–8 June, held at CAiSE 2010, Hammamet, Tunisia. Revised Selected Papers. Berlin: Springer. LNBIP 62, 90–106.
- García, J., García-Peñalvo, F., and Therón, R., 2011a. Modelling relationships among classes as semantic coupling in OWL ontologies. In: *Proceedings of the 2011 international conference on information & knowledge engineering, IKE 2011*, 18–21 July, Las Vegas, Nevada. Vol. 1. Las Vegas Nevada: CSREA Press, 22–28.
- García, J., Therón, R., and García-Peñalvo, F., 2011b. Semantic zoom: details on demand visualisation technique for modelling OWL ontologies. In: *Highlights in practical applications of agents and multiagent systems, 9th international conference on practical applications of agents and multiagent systems – PAAMS 2011*, 6–8 April, Salamanca, Spain. Berlin: Springer. AISC 89, 85–92.
- García, J., et al., 2011c. Usability evaluation of a visual modelling tool for owl ontologies. *Journal of Universal Computer Science* 17 (9), 1299–1313.
- García-Crespo, A., et al., 2010. Semo: a framework for customer social networks analysis based on semantics. *Journal of Information Technology* 25 (2), 178–188.
- García-Peñalvo, F., García, J., and Therón, R., 2011. Analysis of the owl ontologies: A survey. *Scientific Research and Essays* 6 (20), 4318–4329.
- García-Peñalvo, F.J., et al., 2012. Towards an ontology modeling tool. A validation in software engineering scenarios. *Expert Systems with Applications* 39 (13), 11468–11478.
- Gómez-Pérez, A., Fernández, M., and Corcho, O., 2003. *Ontological engineering*. Berlin: Springer-Verlag.
- Gruber, T.R., 1993. Toward principles for the design of ontologies used for knowledge sharing. *Journal International Journal of Human-Computer Studies* 43 (5–6), 907–928.
- Guarino, N. and Giarretta, P., 1995. Ontologies and knowledge bases. *Knowledge Acquisition*, 1, 25–32.
- Holten, D., 2006. Hierarchical edge bundles: visualization of adjacency relations in hierarchical data. *IEEE Transactions on Visualization and Computer Graphics* 12 (5), 741–748.
- Horridge, M., et al., 2011. *A practical guide to building OWL ontologies using the Protégé-OWL plugin and CO-ODE tools Edition 1.3* [online], The University of Manchester. Available from: <http://owl.cs.manchester.ac.uk/tutorials/protegeowltutorial/> [Accessed 1 August 2012].

- Jarrar, M. and Meersman, R., 2002. Formal ontology engineering in the dogma approach. In: *Proceedings of the international conference on ontologies, databases, and applications of semantics (ODBase 2002)*, 29–31 October, Irvine, California. Berlin: Springer. LNCS 2519, 1238–1254.
- Jarrar, M. and Meersman, R., 2008. Ontology engineering – the dogma approach. *Advances in Web Semantics I* 4891, 7–34.
- Johnson, B. and Shneiderman, B., 1991. Treemaps: a space filling approach to the visualization of hierarchical information structures. In: *Proceedings of the 2nd conference on visualization '91 (VIS'91)*, 22–25 October, San Diego, CA. Los Alamitos, CA: IEEE Computer Society Press, 284–291.
- Karlsson, E.-A., 1995. *Software reuse: A holistic approach*. New York, NY: John Wiley and Sons Ltd.
- Keim, D., et al., 2006. Monitoring network traffic with radial traffic analyzer. In: *2006 IEEE symposium on visual analytics science and technology (VAST 2006)*, 31 October–2 November, Baltimore, Maryland. Los Alamitos, CA: IEEE Computer Society Press, 123–128.
- Keim, D., et al., 2008. Visual analytics: combining automated discovery with interactive visualizations. In: *Discovery Science. Proceedings of the 11th international conference, DS 2008*, 13–16 October, Budapest, Hungary. Berlin: Springer-Verlag, LNCS 5255, 2–14.
- Keim, D., et al., 2010. *Mastering the information age solving problems with visual analytics*. Goslar, Germany: Eurographics Association.
- Kerren, A., et al., 2008. *Information visualization – human-centered issues and perspectives*. Berlin: Springer-Verlag.
- Knublauch, H., et al., 2004. The Protégé OWL plugin: an open development environment for semantic web applications. In: *Third international semantic web conference ISWC 2004*, 7–11 November, Hiroshima, Japan. Berlin: Springer. LNCS 3298, 229–243.
- Kosara, R., Miksch, S., and Hauser, H., 2002. Focus+context taken literally. *IEEE Computer Graphics and Applications* 22 (1), 22–29.
- Lintern, R. and Storey, M., 2005. Jambalaya express: ontology visualization-on demand. In: *Proceedings of the 8th international Protégé conference*, 18–21 July, Madrid, Spain. Stanford: Stanford University School of Medicine, 1–3.
- Marcondes, C. and Rocha, M., 2009. Ontological and conceptual bases for a scientific knowledge model in biomedical articles. *Electronic Journal of Communication, Information & Innovation in Health* 3, 19–30.
- Marcondes, C., et al., 2008. Ontologies as the new bases of scientific knowledge. *Perspectivas em Ciencia da Information* 13, 20 – 39.
- Moody, D., 2009. The physics of notations: toward a scientific basis for constructing visual notations in software engineering. *IEEE Transactions on Software Engineering* 35, 756–779.
- PrietoDiaz, R. and Arango, G., 1991. *Domain analysis and software systems modeling*. Los Alamitos, CA: IEEE Computer Society Press.
- Spyns, P., Meersman, R., and Jarrar, M., 2002. Data modelling versus ontology engineering. *SIGMOD Record*, 31, 12–17.
- Stevens, R., et al., 2007. Using owl to model biological knowledge. *International Journal of Human-Computer Studies* 65 (7), 583–594.
- Storey, M., et al., 2001. Jambalaya: Interactive visualization to enhance ontology authoring and knowledge acquisition in Protégé. In: *Proceedings of the workshop on interactive tools for knowledge capture, K-CAP-2001* [online], 20 October, Victoria, BC, Canada. Available from: <http://www.isi.edu/~blythe/kcap-interaction/papers.html>.
- Thomas, J. and Cook, K., 2005. *Illuminating the path: the research and development agenda for visual analytics*. USA: National Visualization and Analytics Center.
- Zhao J.Y., Ordóñez de Pablos, P., Qi, Z., 2012. Enterprise knowledge management model based on China's practice and case study. *Computers in Human Behavior*, 28(2): 324–330.

Copyright of Behaviour & Information Technology is the property of Taylor & Francis Ltd and its content may not be copied or emailed to multiple sites or posted to a listserv without the copyright holder's express written permission. However, users may print, download, or email articles for individual use.